We claim:

1. A method of discovering one or more patterns in two sequences of symbols S_1 and S_2, the symbols being members of an alphabet, the method comprising the steps of:

a) for each sequence, forming a master offset table that groups for each symbol the position in the sequence occupied by each occurrence of that symbol;

b) determining the difference in position between each occurrence of a symbol in one of the sequences and each occurrence of that same symbol in the other sequence;

c) forming a Pattern Map setting forth, for each value of a difference in position, the position in the first sequence of each symbol therein that appears in the second sequence at that difference-in-position value,

the collection of the symbols tabulated for each value of difference in position thereby defining a parent pattern in the first sequence that is repeated in the second sequence.

2. The method of claim 1 further comprising the step of:

d) identifying in the Pattern Map each value of a difference in position wherein the number of symbols tabulated is greater than a predetermined threshold.

3. The method of claim 1 further comprising the steps of:

d) forming a table of ordered (symbol, position) pairs, where position refers to the position of the symbol in a sequence;

e) using the table of ordered pairs, defining the parent pattern by populating the parent pattern with symbols at relative locations indicated by the position of the symbol in the sequence.

4. The method of claim 1 further comprising the step of:

d) defining the parent pattern by populating the parent pattern with symbols at relative locations indicated by the position of the symbol in the sequence.

5. The method of claim 1 wherein each symbol has a position index associated therewith denoting the position of the symbol within the sequence, the method further comprising the steps of:

d) merging a predetermined number of adjacent rows in the Pattern Map and sorting by position indices to create a merge-sorted list, wherein the predetermined number of adjacent rows defines a maximum number of insertions or deletions per location within a pattern;

e) converting each (symbol, position index) pair in the merge-sorted list to a (symbol, position index, total index) triple, where the total index is defined by the sum of the position index and the row index;

f) defining a reference pattern by placing a unique instance of each symbol for a given position index in the merge-sorted list into a first output array at a relative location indicated by the position index;

g) defining a corrupted pattern relative to the reference pattern by placing a unique instance of each symbol for a given position index in the merge-sorted list into a second output array at a relative location indicated by the total index;

h) repeating step f) until all corrupted patterns have been defined, or until a predetermined condition has been reached.

6. The method of claim 1 further comprising the step of:

d) replacing one of the sequences S_1 and S_2 by the parent pattern; and

e) repeating steps a) - c) to discover child patterns in the other sequence.

7. The method of claim 3 further comprising the step of:

f) specifying a predetermined maximum allowable number of placeholders in the pattern,

g) specifying a predetermined minimum number of symbols for a candidate pattern;

h) defining a candidate pattern from the patent pattern by selecting an interval in the parent pattern, the interval containing the predetermined maximum number of placeholders; and

i) comparing the number of symbols included in the candidate pattern with the predetermined minimum number of symbols.

8. The method of claim 3 further comprising the step of:

f) specifying a predetermined maximum allowable number of contiguous placeholders between symbols in the pattern,

g) specifying a predetermined minimum number of symbols for a candidate pattern;

h) defining a candidate pattern from the patent pattern by selecting an interval in the parent pattern, the interval containing the predetermined maximum allowable number of contiguous placeholders; and

i) comparing the number of symbols included in the candidate pattern with the predetermined minimum number of symbols.

9. The method of claim 1 wherein the first sequence of symbols has a length L1 and the second sequence of symbols has a length L2, and wherein each symbol has a position index associated therewith

denoting the position of the symbol within the sequence, further comprising the step of:

prior to step a), adjusting the position index of each symbol in the second sequence by adding L1 to each position index thereof to define the position of each symbol relative to the start of the first sequence.

10. The method of claim 1 further comprising the step of:

prior to step a), translating the sequences from the alphabet in which they were originally written to an alternate alphabet by a predetermined mapping function.

11. A method of discovering one or more patterns in two sequences of symbols, the symbols being members of an alphabet, the first sequence of symbols having a length L1 and the second sequence of symbols having a length L2, comprising the steps of:

a) translating the sequences of symbols into a table of ordered (symbol, position index) pairs, where the position index of each (symbol, position index) pair refers to the location of the symbol in a sequence;

b) for each of the two sequences, grouping the (symbol, position index) pairs by symbol to respectively form a first master offset table and a second master offset table;

c) forming a Pattern Map comprising an array having (L1 + L2 -1) rows by:

i) subtracting the position index of the first master offset table from the position index of the second master offset table for every combination of (symbol, position index) pair having like symbols, the difference resulting from each subtraction defining a row index;

ii) repeatedly storing each (symbol, position index) pair from the first master offset table in

a row of the Pattern Map, the row being defined by the row index, until all (symbol, position index) pairs have been stored in the Pattern Map;

d) defining a parent pattern by populating an output array with the symbols of each (symbol, position index) pair of a row of the Pattern Map, the symbols being placed at relative locations in the parent pattern indicated by the position index of the pair; and

e) repeating step d) for each row of the Pattern Map.

12. The method of claim 11 further comprising the step of:

prior to step a), adjusting the position index of each symbol in the second sequence is adjusted by adding L1 to the position index to define the position of each symbol relative to the start of the first sequence, thereby to concatenate the sequences.

13. The method of claim 11 wherein step c) further comprises:

iii) sorting each row of the Pattern Map by the indices of the (symbol, position index) pairs stored therein.

14. The method of claim 11 wherein the first sequence and the second sequence are the same.

15. The method of claim 11 for discovering one or more patterns in a set of sequences of symbols, further comprising the steps of:

f) repeating steps a) - e) for all possible pairwise combinations of sequences in the set of sequences to define a set of parent patterns.

16. The method of claim 15 further comprising the steps of:

g) translating each pattern into an array of ordered pairs of (symbol, position index), sorted in position index order;

h) specifying a predetermined maximum allowable number of placeholders, and specifying a predetermined minimum number of symbols for a candidate pattern;

i) setting an interval start pointer to the first ordered pair of the pattern;

j) extending an interval of ordered pairs to the right of the pointer until the number of placeholders exceeds the predetermined maximum number of placeholders, said interval defining a candidate pattern;

k) if the number of symbols included in the candidate pattern is greater than or equal to the predetermined minimum number of symbols, designating the candidate pattern of step j) as a trimmed pattern and outputting the trimmed pattern to an output table;

l) advancing the start point to the next ordered pair; and

m) repeating steps j) and k) until the number of symbols remaining beyond the pointer in the array of ordered pairs is less than the predetermined minimum number of symbols, to define a set of trimmed patterns.

17. The method of claim 15 further comprising the steps of:

g) translating each pattern into an array of ordered pairs of (symbol, position index), sorted in position index order;

h) specifying a predetermined gap size g, and specifying a predetermined minimum number of symbols n;

i) setting a pointer at the beginning of the array of ordered pairs;

j) repeatedly calculating a gap between adjacent ordered pairs in the array by finding the position index difference between the indices of adjacent

ordered pairs until a gap greater than or equal to the gap size g is found, the symbols between the pointer and the gap defining a candidate pattern;

k) if the number of symbols between the pointer and the beginning of the gap is greater than or equal to the minimum number of symbols n, designating the candidate pattern as a chopped pattern and outputting the chopped pattern to an output table;

l) moving the pointer to the ordered pair immediately after the current gap, and

m) repeating steps j) - k) until the number of symbols remaining beyond the pointer is less than the minimum number of symbols n to define a set of chopped patterns.

18. The method of claim 11 further comprising:

f) deleting all patterns not satisfying a predetermined criteria.

19. The method of claim 11 further comprising:

f) deleting all patterns shorter than a first predetermined span and longer than a second predetermined span.

20. The method of claim 11 further comprising: f) deleting all patterns having fewer than a predetermined number of symbols.

21. A method of discovering one or more corrupted patterns, relative to a reference pattern, in two sequences of symbols, the corrupted patterns being corrupted by insertions and/or deletions, comprising the steps of:

a) translating the sequences of symbols into a table of ordered (symbol, position index) pairs, where the position index refers to the location of the symbol in the one or more sequences;

b) for each of the two sequences, grouping the

(symbol, position index) pairs by symbol to respectively form a first master offset table and a second master offset table;

c) forming a Pattern Map comprising an array having (L1 + L2 -1) rows by:

i) subtracting the position index of the first master offset table from the position index of the second master offset table for every combination of (symbol, position index) pair having like symbols, the difference resulting from each subtraction defining a row index;

ii) repeatedly storing each (symbol, position index) pair from the first master offset table in a row of the Pattern Map, the row being defined by the row index, until all (symbol, position index) pairs have been stored in the Pattern Map;

iii) merging a predetermined number of adjacent rows and sorting by the position indices stored therein to create a merge-sorted list, wherein the predetermined number of adjacent rows defines a maximum number of insertions or deletions per location within a pattern;

iv) converting each (symbol, position index) pair in the merge-sorted list to a (symbol, position index, total index) triple, where the total index is defined by the sum of the position index and the row index;

d) defining a reference pattern by placing a unique instance of each symbol for a given position index in the merge-sorted list into a first output array at a relative location indicated by the position index;

e) defining a corrupted pattern relative to the reference pattern by placing a unique instance of each symbol for a given position index in the merge-sorted

list into a second output array at a relative location indicated by the total index; and

f) repeating step e) until all corrupted patterns have been defined, or until a predetermined condition has been reached.

22. The method of claim 21 where the number of corrupted patterns is the product of the number of times each position index occurs within the merge-sorted list.

23. The method of claim 21 where the predetermined condition of step f) is a predetermined cumulative number of symbol insertions or symbol deletions permitted in the corrupted patterns.

24. The method of claim 21 where the repeating process in step f) is performed recursively on the merge-sorted list.

25. The method of claim 21 further comprising:

g) deleting all corrupted patterns not satisfying a predetermined criteria.

26. The method of claim 21 further comprising the step of, prior to step a), translating the sequences from the alphabet in which they were originally written to an alternate alphabet by a predetermined mapping function.

27. The method of claim 26 for discovering patterns corrupted by substitutions, wherein the alternate alphabet has fewer symbols than the original alphabet and the mapping is such that one or more symbols of the original alphabet map to one symbol of the alternate alphabet.

28. The method of claim 26 wherein the alternate alphabet has the same number of symbols or more symbols than the original alphabet.

29. A method of discovery of child patterns of increased support in two sequences of symbols, the symbols being members of an alphabet, the first sequence of symbols having a length L1 and the second sequence of symbols having a length L2, comprising the steps of:

a) translating the sequences of symbols into a table of ordered (symbol, position index) pairs, where the position index of each (symbol, position index) pair refers to the location of the symbol in a sequence;

b) for each of the two sequences, grouping the (symbol, position index) pairs by symbol to respectively form a first master offset table and a second master offset table;

c) forming a Pattern Map comprising an array having (L1 + L2 -1) rows by:

i) subtracting the position index of the first master offset table from the position index of the second master offset table for every combination of (symbol, position index) pair having like symbols, the difference resulting from each subtraction defining a row index;

ii) repeatedly storing each (symbol, position index) pair from the first master offset table in a row of the Pattern Map, the row being defined by the row index, until all (symbol, position index) pairs have been stored in the Pattern Map;

d) defining a parent pattern by populating an output array with the symbols of each (symbol, position index) pair of a row of the Pattern Map, the symbols being placed at relative locations in the parent pattern indicated by the position index of the pair; and

e) repeating step d) for each row of the Pattern Map;

f) repeating steps a - e for all possible pairwise combinations of sequences in the set of sequences to define a set of parent patterns;

g) replacing the second sequence by a parent pattern previously discovered in step f); and

h) repeating steps a) - f) to discover child patterns in the first sequence.

30. The method of claim 29 further comprising:

i) repeating steps g) and h) for each parent pattern previously discovered in step f); and

j) repeating steps a) - f) to discover all child patterns in the first sequence.

31. The method of claim 30 further comprising:

k) replacing the first sequence by a parent pattern previously discovered in step f); and

l) repeating steps a) - f) to discover child patterns in the second sequence.

32. The method of claim 31 further comprising:

m) repeating steps k) and l) for each parent pattern previously discovered in step f); and

n) repeating steps a) - f) to discover all child patterns in the second sequence.

33. A method of discovery of child patterns of increased support in two sequences of symbols, the symbols being members of an alphabet, the first sequence of symbols having a length L1 and the second sequence of symbols having a length L2, comprising the steps of:

a) translating the sequences of symbols into a table of ordered (symbol, position index) pairs, where the position index of each (symbol, position index) pair refers to the location of the symbol in a

sequence;

b) for each of the two sequences, grouping the (symbol, position index) pairs by symbol to respectively form a first master offset table and a second master offset table;

c) forming a Pattern Map comprising an array having (L1 + L2 -1) rows by:

i) subtracting the position index of the first master offset table from the position index of the second master offset table for every combination of (symbol, position index) pair having like symbols, the difference resulting from each subtraction defining a row index;

ii) repeatedly storing each (symbol, position index) pair from the first master offset table in a row of the Pattern Map, the row being defined by the row index, until all (symbol, position index) pairs have been stored in the Pattern Map;

d) defining a parent pattern by populating an output array with the symbols of each (symbol, position index) pair of a row of the Pattern Map, the symbols being placed at relative locations in the parent pattern indicated by the position index of the pair; and

e) repeating step d) for each row of the Pattern Map;

f) repeating steps a - e for all possible pairwise combinations of sequences in the set of sequences to define a set of parent patterns;

g) replacing the first sequence by a first parent pattern previously discovered in step f) and replacing the second sequence by a second parent pattern previously discovered in step f); and

h) repeating steps a) - f) to discover child patterns in the first parent pattern.

34. The method of claim 33 further comprising:

i) repeating steps g) and h) for all pairwise combinations of first and second parent patterns previously discovered in step f); and

j) repeating steps a) - f) to discover all child patterns in the parent patterns.

35. A method of discovering one or more patterns in a set of k sequences of symbols, called a k-tuple, where k is greater than or equal to two, within an overall set of w sequences, the symbols being members of an alphabet, each sequence of symbols having respective lengths L_1, L_2, ..., L_w, comprising the steps of:

a) translating the sequences of symbols into a table of ordered (symbol, position index) pairs, where the position index refers to the location of the symbol in a sequence;

b) for each sequence, grouping the (symbol, position index) pairs by symbol to form a respective master offset table, thus creating w master offset tables;

c) forming a k-tuple table associated with the k-tuple, the table comprising k, each column corresponding to one of the k sequences;

i) the first, primary, column comprising the (symbol, position index) pairs of the first, primary, sequence,

ii) the subsequent (k-1) suffix columns comprising (symbol, difference-in-position value) pairs, where the difference-in-position value are the position differences between all possible like symbols of each remaining sequence of the tuple and the primary sequence of the tuple,

iii) the rows in the k-tuple table resulting from forming all possible combinations of like symbols from each sequence;

d) creating a sorted k-tuple table by performing a multi-key sort on the k-tuple table, the sort keys being selected respectively from the difference-in-position value of the last suffix column (k^{th} column) through the difference-in-position value of the first suffix column (2^{nd} column);

e) defining a set of patterns by collecting adjacent rows of the sorted k-tuple table whose suffix columns contain identical sets of difference-in-position values, the relative positions of the symbols in each pattern being determined by the primary column position indices, the set of patterns being common to the k sequences.

36. The method of claim 35 further comprising:

f) deleting all patterns not satisfying a predetermined criteria.

37. The method of claim 35 further comprising:

f) deleting all patterns shorter than a first predetermined span and longer than a second predetermined span.

38. The method of claim 35 further comprising:

f) deleting all patterns having fewer than a predetermined number of symbols.

39. The method of claim 35, further comprising the step of deleting rows from the k-tuple table which do not have suffix indices identical to any other row of the k-tuple table.

40. The method of claim 35 further comprising the step of deleting rows from the k-tuple table according to predetermined criteria.

41. The method of claim 40, wherein rows are deleted from the k-tuple table if there are fewer than

N_s rows sharing identical suffix column difference-in-position values, where N_s is the minimum number of symbols per pattern.

42. A method of forming a (k+1)-tuple table, wherein a k-tuple table is combined with a sequence, comprising the steps of:

a) translating the sequence of symbols into a table of ordered (symbol, position index) pairs, where the position index refers to the location of the symbol in a sequence;

b) grouping the (symbol, position index) pairs by symbol to form a respective master offset table;

c) creating the (k+1)-tuple table of k+1 columns by:

i) forming all combinations of like symbols between the primary column of the k-tuple table and the master offset table,

ii) for each such combination, duplicating the corresponding row of the k-tuple table, and appending a (symbol, difference-in-position value) pair corresponding to the difference between the position index of the master offset table and the position index of the primary column.

43. The method of claim 42 further comprising the step of:

deleting patterns from a k-tuple table common to the k-tuple table and a (k+1)-tuple table, where the (k+1)-tuple table contains all of the sequences of the k-tuple table with one addition sequence, by:

a) deleting the suffix column corresponding to a sequence not shared between the two tuple tables, thereby defining a modified table, and

b) deleting rows from the k-tuple table whose suffix columns contain identical sets of

difference- in-position values to a row of the modified table.

44. A method of discovering one or more patterns in a set of k sequences of symbols, called a k-tuple, comprising the steps of:

a) for a first two sequences of the k-tuple

i) translating each sequence of symbols into a table of ordered (symbol, position index) pairs, where the position index of each (symbol, position index) pair refers to the location of the symbol in the sequence;

ii) for each of the two sequences, grouping the (symbol, position index) pairs by symbol to respectively form a first master offset table and a second master offset table;

iii) forming a Pattern Map comprising an array having (L1 + L2 -1) rows by:

A) subtracting the position index of the first master offset table from the position index of the second master offset table for every combination of (symbol, position index) pair having like symbols, the difference resulting from each subtraction defining a row index;

B) repeatedly storing each (symbol, position index) pair from the first master offset table in a row of the Pattern Map, the row being defined by the row index, until all (symbol, position index) pairs have been stored in the Pattern Map;

iv) defining a parent pattern by populating an output array with the symbols

of each (symbol, position index) pair of a row of the Pattern Map, the symbols being placed at relative locations in the parent pattern indicated by the position index of the pair; and

v) repeating step d) for each row of the Pattern Map;

b) storing the discovered patterns as arrays of (symbol, position index) pairs;

c) for each subsequent sequence of the k-tuple, replacing the (symbol, position index) pairs of the first sequence by the (symbol, position index) pairs of the stored patterns; and

d) repeating steps (a) through (c) until level k of the k-tuple is reached.

45. The method of claim 35, wherein the method of finding all patterns at all levels of support from a set of sequences comprises the steps of:

a) forming a tree of nodes, where each node corresponds to each possible combination of sequences in an ordered set of sequences, and also therefore to a corresponding k-tuple;

b) organizing the nodes into a tree structure, wherein a node with a k-tuple is connected to all possible nodes containing (k+1) tuples, the (k+1) tuple being formed by adding a unique sequence to the k-tuple, where the sequence being added is later in the ordered list of sequences than the latest sequence of the k-tuple;

c) traversing the tree, and at each node visited during traversal, defining a set of patterns by collecting adjacent rows of the sorted k-tuple table whose suffix columns contain identical sets of difference-in-position values, the relative positions of the symbols in each pattern being determined by the primary column position indices, the set of patterns being common to the k sequences.

46. The method of claim 45, wherein the traversal of the tree of nodes is accomplished via recursion.

47. The method of claim 45, further comprising the step of:

d) removing duplicate patterns at each level of support.

48. The method of claim 47, wherein the removal of duplicate patterns is accomplished by:

i) for each node corresponding to a (k+1)-tuple, identifying the nodes containing k-tuples whose sequences are subsets of the (k+1)-tuple; thereby defining a set of causally-dependent nodes;

ii) locating said causally-dependent nodes;

iii) removing from each said causally-dependent node the patterns in common with the (k+1)-tuple; and

iv) if the k-tuple table in a causally-dependent node is thereby reduced to zero length, removing the corresponding node and all of its descendents from the tree prior to their traversal.

49. The method of claim 48, wherein locating causally-dependent nodes in step ii) comprises the steps of:

(a) organizing the nodes at level k in the Tuple-tree into a linked list which is ordered from left to right in accordance with the sequence numbers of each tuple; and

(b) searching said linked list for nodes which are causally-dependent on a particular (k+1)-tuple.

50. The method of claim 48, wherein the nodes located in step ii) are causally-dependent nodes at level k determined with respect to another node at level k, and are thus causally-dependent on a child of the another node at level k.

51. The method of claim 47, wherein the removal of duplicate patterns at each level of support comprises the steps of:

i) organizing the nodes at level k in the Tuple-tree into a linked list which is ordered from left to right in accordance with the sequence numbers of each tuple;

ii) for each pattern in the current node at level k, storing a "hit list" array of the indices of the sequences containing the pattern;

iii) for all nodes to the right of the current node whose indices are all in the hit list, searching for a duplicate instance of the pattern, and if found, eliminating it; and

iv) making each node the current node, repeating steps (ii) and (iii), in the order of the node's appearance in the linked list.

52. The method of claim 51, wherein, in step iii), the nodes consistent with the hit list are found using a hash tree, the hash tree having a root and k levels of nodes, the k-th level of the hash tree having a plurality of leaf nodes, the respective level of nodes of the hash tree corresponding to the respective index of a k-tuple, the leaf nodes identifying the k-tuple whose indices correspond to the path from the root to the leaf node, wherein

searching the nodes for pattern duplicates is performed by repeating steps ii) and iii) for each node in the order of the appearance of that node in the hash tree.

53. The method of claim 45 wherein the traversing step c) itself comprises the steps of:

i) finding a P-node list corresponding to the location of the pattern in the primary sequence of that tree node,

ii) searching the P-node list for a duplicate instance of the pattern,

(a) if no duplicate is found:

(i) adding a pointer to the pattern to the current T-node pattern array,

(ii) finding all locations of the pattern within the Virtual Sequence Array,

(iii) adding a pointer to the pattern to each corresponding P-node array;

(b) if a duplicate pattern is found:

(i) ignoring the pattern if the duplicate pattern was found at support equal to the current level of support,

(ii) if the duplicate pattern was found at a previous level of support, unlinking the duplicate pattern from its previous T-node (if it exists), and relinking the duplicate pattern to the current T-node,

iii) repeating steps i) and ii) until all of the children of a T-node have been created, thus insuring that patterns on that T-node that are at their ultimate level of support are reported, and

iv) deleting the T-node.

54. A method of discovering all patterns at all levels of support, comprising the steps of :

a) creating a Virtual Sequence Array comprising an array of length equal to the sum of the lengths of all sequences in a set of sequences, each element the array being a pointer to a P-node list that corresponds to

the unique position of each symbol in each sequence, the P-node list being a doubly-linked list of pointers to pattern data structures containing patterns, wherein each pattern data structure contains:

(a) an array of (symbol, position index) pairs describing the pattern;

(b) a set of values describing characteristics of the pattern;

(c) a set of reciprocal pointers to the corresponding P-nodes; and

(d) a pointer to a node in the tuple-tree, called a T-node, the T-node containing information locating the node within the tuple-tree and containing an array of pointers to the patterns belonging to that node.

55. A computer-readable medium containing instructions for controlling a computer system to discover one or more patterns in two sequences of symbols S_1 and S_2, the symbols being members of an alphabet, by performing the steps of:

a) for each sequence, forming a master offset table that groups for each symbol the position in the sequence occupied by each occurrence of that symbol;

b) determining the difference in position between each occurrence of a symbol in one of the sequences and each occurrence of that same symbol in the other sequence;

c) forming a Pattern Map setting forth, for each value of a difference in position, the position in the first sequence of each symbol therein that appears in the second sequence at that difference-in-position value,

the collection of the symbols tabulated for each value of difference in position thereby defining a parent pattern in the first sequence that is repeated in the second sequence.

56. The computer-readable medium of claim 55 further containing instructions for controlling a computer system to perform the step of:

d) identifying in the Pattern Map each value of a difference in position wherein the number of symbols tabulated is greater than a predetermined threshold.

57. The computer-readable medium of claim 55 further containing instructions for controlling a computer system to perform the steps of:

d) forming a table of ordered (symbol, position) pairs, where position refers to the position of the symbol in a sequence;

e) using the table of ordered pairs, defining the parent pattern by populating the parent pattern with symbols at relative locations indicated by the position of the symbol in the sequence.

58. The computer-readable medium of claim 55 further containing instructions for controlling a computer system to perform the step of:

d) defining the parent pattern by populating the parent pattern with symbols at relative locations indicated by the position of the symbol in the sequence.

59. The computer-readable medium of claim 55 wherein each symbol has a position index associated therewith denoting the position of the symbol within the sequence, the medium further containing instructions for controlling a computer system to perform the steps of:

d) merging a predetermined number of adjacent rows in the Pattern Map and sorting by position indices to create a merge-sorted list, wherein the predetermined number of adjacent rows defines a maximum number of insertions or deletions per location within a pattern;

e) converting each (symbol, position index) pair in the merge-sorted list to a (symbol, position index, total index) triple, where the total index is defined by the sum of the position index and the row index;

f) defining a reference pattern by placing a unique instance of each symbol for a given position index in the merge-sorted list into a first output array at a relative location indicated by the position index;

g) defining a corrupted pattern relative to the reference pattern by placing a unique instance of each symbol for a given position index in the merge-sorted list into a second output array at a relative location indicated by the total index;

h) repeating step f) until all corrupted patterns have been defined, or until a predetermined condition has been reached.

60. The computer-readable medium of claim 55 further containing instructions for controlling a computer system to perform the steps of:

d) replacing one of the sequences S_1 and S_2 by the parent pattern; and

e) repeating steps a) - c) to discover child patterns in the other sequence.

61. The computer-readable medium of claim 57 further containing instructions for controlling a computer system to perform the steps of:

f) specifying a predetermined maximum allowable number of placeholders in the pattern,

g) specifying a predetermined minimum number of symbols for a candidate pattern;

h) defining a candidate pattern from the patent pattern by selecting an interval in the parent pattern, the interval containing the predetermined maximum number of placeholders; and

i) comparing the number of symbols included in the candidate pattern with the predetermined minimum number of symbols.

62. The computer-readable medium of claim 57 further containing instructions for controlling a computer system to perform the steps of:

f) specifying a predetermined maximum allowable number of contiguous placeholders between symbols in the pattern,

g) specifying a predetermined minimum number of symbols for a candidate pattern;

h) defining a candidate pattern from the patent pattern by selecting an interval in the parent pattern, the interval containing the predetermined maximum allowable number of contiguous placeholders; and

i) comparing the number of symbols included in the candidate pattern with the predetermined minimum number of symbols.

63. The computer-readable medium of claim 55 wherein the first sequence of symbols has a length L1 and the second sequence of symbols has a length L2, and wherein each symbol has a position index associated therewith denoting the position of the symbol within the sequence, the medium further containing instructions for controlling a computer system to perform the step of:

prior to step a), adjusting the position index of each symbol in the second sequence by adding L1 to each position index thereof to define the position of each symbol relative to the start of the first sequence.

64. The computer-readable medium of claim 55 further containing instructions for controlling a computer system to perform the step of:

prior to step a), translating the sequences from the alphabet in which they were originally written to

an alternate alphabet by a predetermined mapping function.

65. A computer-readable medium containing a data structure useful in controlling a computer system to discover one or more patterns in two sequences of symbols,

the data structure grouping, for each symbol, the position (position index) in the sequence occupied by each occurrence of that symbol.

66. A computer-readable medium containing a data structure useful in controlling a computer system to discover one or more patterns in two sequences of symbols, the data structure grouping,

for each value of a difference in position between each occurrence of a symbol in one of the sequences and each occurrence of that same symbol in the other sequence,

the position (position index) in the first sequence of each symbol therein that appears in the second sequence at that difference-in-position value.

67. The computer-readable medium of claim 66 wherein the data structure further groups, for each value of a difference in position, an indication of the number of symbols in the first sequence that appear in the second sequence at that difference-in-position value.